

IN THE CLAIMS:

1. (Previously Presented) A client-side method of selecting a proxy server storing a web resource from among a plurality of proxy servers, said method comprising the steps of:
5 receiving at said client a request for said web resource;
determining if said web resource is a heavy file type, wherein a given file type is determined to be said heavy file type if said given file type satisfies one or more predefined criteria based on a size of files of said given file type; and
10 redirecting by said client said web resource request to a proxy server associated with said heavy file type when it is determined that said web resource is said heavy file type.
2. (Previously Presented) The method according to claim 1, wherein said heavy file type has an average size that exceeds a predefined threshold.
- 15 3. (Original) The method according to claim 1, wherein said redirecting step further comprises the step of accessing a proxy selection table that associates said file type to a proxy server.
- 20 4. (Original) The method according to claim 1, wherein said redirecting step further comprises the step of redirecting said request to a given proxy server based on the recent history of client request patterns.
5. (Original) The method according to claim 1, further comprising the step of analyzing the recent history of client request patterns.
25
6. (Previously Presented) The method according to claim 1, further comprising the step of assigning $P \times 1/h$ of the available proxy servers to serve heavy file types, where P is the total number of proxy servers and the heavy file types account for a fraction $1/h$ of a total load.

7. (Previously Presented) A client-side method of selecting a proxy server storing a web resource from among a plurality of proxy servers, said method comprising the steps of:

receiving at said client a request for said web resource;

5 determining if said web resource request is served by a domain having a traffic volume that satisfies one or more predefined criteria; and

redirecting by said client said web resource request to a proxy server associated with said domain when it is determined that said web resource request is served by a domain having a traffic volume that satisfies said one or more predefined criteria.

10

8. (Previously Presented) The method according to claim 7, wherein said predefined criteria is based on a maximum normalized daily load.

9. (Original) The method according to claim 7, wherein said redirecting step further
15 comprises the step of accessing a proxy selection table that associates said domain to a proxy server.

10. (Original) The method according to claim 7, wherein said redirecting step further comprises the step of redirecting said request to a given proxy server based on the recent history of client request patterns.

20

11. (Original) The method according to claim 7, further comprising the step of analyzing the recent history of client request patterns.

12. (Previously Presented) The method according to claim 7, further comprising the steps
25 of sorting heavy domains in increasing order of their average file sizes, splitting said sorted list into $P \times (1 - (1/h))$ partitions of equal load, and assigning one partition to each of the remaining proxy servers, where P is the total number of proxy servers and the heavy file types account for a fraction $1/h$ of the total load.

13. (Previously Presented) A client-side system for selecting a proxy server storing a web resource from among a plurality of proxy servers, said system comprising:

a memory for storing computer readable code; and

a processor operatively coupled to said memory, said processor configured to:

receive at said client a request for said web resource;

determine if said web resource is a heavy file type, wherein a given file type is determined to be said heavy file type if said given file type satisfies one or more predefined criteria based on a size of files of said given file type; and

redirect by said client said web resource request to a proxy server associated with said heavy file type when it is determined that said web resource is said heavy file type.

14. (Previously Presented) The system according to claim 13, wherein said heavy file type has an average size that exceeds a predefined threshold.

15. (Original) The system according to claim 13, wherein said memory further includes a proxy selection table that associates said file type to a proxy server.

16. (Original) The system according to claim 13, wherein said processor is further configured to redirect said request to a given proxy server based on the recent history of client request patterns.

17. (Previously Presented) A client-side system for selecting a proxy server storing a web resource from among a plurality of proxy servers, said system comprising:

a memory for storing computer readable code; and

a processor operatively coupled to said memory, said processor configured to:

receive at said client a request for said web resource;

determine if said web resource request is served by a domain having a traffic volume

that satisfies one or more predefined criteria; and

redirect by said client said web resource request to a proxy server associated with said domain when it is determined that said web resource request is served by a domain having a traffic volume that satisfies said one or more predefined criteria.

5

18. (Previously Presented) The system according to claim 17, wherein said predefined criteria is based on a maximum normalized daily load.

19. (Original) The system according to claim 17, wherein said memory further includes a proxy selection table that associates said domain to a proxy server.

10

20. (Original) The system according to claim 17, wherein said processor is further configured to redirect said request to a given proxy server based on the recent history of client request patterns.

15

21. (Previously Presented) An article of manufacture for client-side selection of a proxy server storing a web resource from among a plurality of proxy servers, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

20

a step to receive at said client a request for said web resource;

a step to determine if said web resource is a heavy file type, wherein a given file type is determined to be said heavy file type if said given file type satisfies one or more predefined criteria based on a size of files of said given file type; and

25

a step to redirect by said client said web resource request to a proxy server associated with said heavy file type when it is determined that said web resource is said heavy file type.

22. (Previously Presented) An article of manufacture for client-side selection of a proxy server storing a web resource from among a plurality of proxy servers, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to receive at said client a request for said web resource;

5 a step to determine if said web resource request is served by a domain having a traffic volume that satisfies one or more predefined criteria; and

a step to redirect by said client said web resource request to a proxy server associated with said domain when it is determined that said web resource request is served by a domain having a traffic volume that satisfies said one or more predefined criteria.